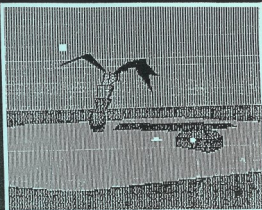


NEWSLETTER AUG/SEPT 1992  
Issue No. 8



3D

CONSTRUCTION  
KIT USERS GROUP

## EDITORIAL

Welcome to issue 8 of the 3D Construction Kit User Group Newsletter.

Thanks to everyone who sent in contributions for the Newsletters following my plea in the last issue. I was intending to make this a real bumper issue with loads more extra pages but I think you will fully understand my reasons for not doing so when I tell you that my workload increased while I was typing this all up - and in a way that will please the majority of you - I am doing the playtesting on the long-awaited Kit II!! Yes I've finally got my hands on it! I got the distinct impression over the last few months that you are all extremely eager to get hold of this new version so I thought that the sooner I got down to working on it the sooner you would all have it! But, this issue IS bigger than previous issues AND you will have found a bonus in the envelope too! An index of everything that was contained in all the issues to date! Quite a few of you sent in an index, following the plea by Martin Sullivan in the last issue, and I thank you all for your kind efforts. I was only sorry that I couldn't use them all. The index enclosed was kindly provided by Mark Rose and should help you all to make full use of your Newsletters and to find items you were searching for without difficulty. I've also received quite a few new games and routines for our PD library for review but those too will have to wait until the next issue when I will have more time to devote to them. I feel that it is only fair to everyone that I play the games properly and not just give them a cursory glance - especially after all the months or work that go into them.

I am afraid that I cannot release any details about 3D Kit II to you all just yet as it is still under wraps until all the playtesting is complete so you will have to be patient with me. As soon as I have finished work on it and passed it back to the programmers I will be able to tell you all about it. What I can tell you is that I was absolutely delighted and that my flabber has never been so gashed as it was when I saw all the extra incredible features included on it! You are all in for a real treat! You shouldn't have too long to wait now for its release - a lot now depends on how thoroughly and how fast I can do the playtesting so you will understand why I am devoting a lot of my time over the next few weeks to doing the best job I can.

Many of you have written asking if I have the results of the competition for the CD TV yet. Well, to be honest, I do actually know who the outright winner was but until Domark have finally decided on all the objects they are going to include on the disk I cannot publish the complete list of winners so you are going to have to wait a little longer. I hope to have the complete results in time for the next newsletter. Speaking of competitions reminds me to mention that I have only had THREE entries for the 8 Bit competition featured in the last issue!! What's happened to the 8 Bitters? In case anyone else would like to enter the competition I am extending the deadline for entries to 1st October and hope that I will get more entries by then. So the next Newsletter will have the results of both the competitions in it! Also, hopefully, the next issue will contain a complete review of 3D Kit II - all being well that is.

Yikes! The bottom of the page is rapidly approaching once more - I do go on, don't I? Hope you enjoy this issue and find it useful. See you all again with the October/November '92 issue so byeee for now,

*Mandy*

## LETTERS

*Dear Mandy*

I have some questions that I wonder if you can answer. Are the datafiles between the Atari ST and Amiga versions of the same format? would it be possible to transfer the data from one machine to the other by saving from ST to PC compatible disk then copy it to an Amiga disk using a utility like MessyDOS then directly load it into the Amiga version of the Kit? I am pretty keen on the idea of a 3D Kit User convention, it sounds like it could be fun. I would certainly go if it isn't too far away. It would be very useful to members who have problems with the Kit as they could be shown directly by someone with a lot of experience how to get around their troubles.

*Robin Ball, Devizes - ATARI ST*

*I am not very good at the technical side of things, Robin, but I have recently been discussing this problem with Ozzy Omara and Chris King. The problem is mainly with the Amiga as it is extremely fussy about what types of disks it will handle. Chris assures me that the problem can be overcome with the use of a null modem lead. Perhaps Chris or Ozzy would like to write in with news of what they have discovered and between us all we might come up with a solution to the problem...Mandy*

*Dear Mandy*

Here are some ideas I have had about what types of game can be created using the Kit. 3D games tend to fall into several categories such as: Rescue, Sabotage, Treasure Hunting or Escape (many are even a mixture). You have to have limits in what can be put into a particular game setting. For instance you wouldn't put a flying saucer from the planet Zeb in a game set in the Middle Ages now would you - at least I hope not! When you are looking for ideas, try turning to films and other computer games for inspiration. Ideas can also be gleaned by taking a look at the games on offer in the PD library. If you are totally hopeless at animation or variable handling don't despair as many types of games don't need them, eg. if you create a maze all the player has to do is find their way out again so no need for any animations or vexing variables. An interesting idea to try would be to mix various settings so you could end up on an alien planet where the inhabitants build in the same style as the ancient Egyptians. So go now o potential producers of pixel perfect polygon pandemonium and start creating those games!

*Stephen Forsyth, Cumbria. Age 13 - ATARI ST*

*Dear Mandy*

The User Group Game is an excellent idea which has revived my interest in the Kit. This had waned as I found creating the objects rather a struggle and I wish it were possible to use the attributes box to fix the size and position of objects or have an accelerator key to make one key press equal to, say, ten. The most frustrating snag is, however, the unreliability of the saving to disc. So many of my saves fail causing the machine to hang up and I have to reboot the program and data. I have found that my problem with saving to disc is lessened if I save two or three times during the creation, editing and shading of each object. I then have less failures - about 1 in 25 saves, so the amount that the data has changed seems to be a factor. I would appreciate knowing more about the Global Objects, a subject which is not fully covered in the manual so an article on that would be welcome. One suggestion for castle-builders (Amstrad variety), is to replace 09 (green) by 10 (cyan) in the default area colouring. This then gives a nice range of greys for stone castle walls, floors



etc, with bright red and yellow and a pale green for interesting "fixtures and fittings". Having joined in my childrens games of "Hero Quest" I shall have small boxes of gold hidden around, and shall populate the rooms with hostile robots (suits of armour?).

**Humphrey Bolton**, Mirfield - AMSTRAD CPC

Thank goodness it isn't only me who gets roped into games of Hero Quest with the children! (Actually I love it, however, I don't see why I should never be allowed a turn at being "The Evil Wizard" chap who commands everything!) Never mind, Humphrey, the school holidays will soon be over. I am sorry you have had such a frustrating time with the saving of data. When this problem was first pointed out to me I passed the information on to the programmers but I am still waiting to hear if they have found a fix for that. I'll let you know when I hear anything.....Mandy

**Dear Mandy**

Please find enclosed my subscription for a further year of Newsletters and my very best wishes for the next year to everyone. I'm also requesting some PD disks, hopefully I'll get a bit more insight as I am stuck in a rut at the moment. I can't seem to get the hang of the variables, especially when reading about making full use of the 32 bits variables have, and the flags?, things like ORV and ANDV commands, I know that I'm just not grasping all the facts that are staring out at me in the Newsletters, even though the members make it as simple as possible, it's just all those 1111s and numbers and how to allocate conditions (flags?) to the separate bits, it just seems so easy to some people. I'm really glad that you are starting a User Group Game for us members who are still struggling, it is going to be a great incentive for us to try and catch up a little.

**D. Spencer**, Salford - AMIGA

I think I may have confused you even further in my explanations of Variables by mentioning flags. I used the terms "Flags", "Markers", "Counters" etc within my comments as an aid in explaining variables to people who have used other utilities for programming such as PAW, GAC, QUILL, STAC etc, but of course they would be confusing for people who have never come across those utilities. Variables are the "heart" of more complicated conditions used to create problems for the player to solve. For a beginner to try and understand the different bits is a might too complicated so stick to using variables in their most simple form to start with and the rest will come with time. For an example you could start with a simple problem such as finding the rope in the Kitgame and being able to use it to climb down the cliff to the boat only if you have obtained it first from the puzzle zone. The programmers used variable 42 for this and decided that if variable 42 was set to 0 then the rope hadn't been found so the player couldn't get down the cliff but if variable 42 contained the number 1 then the rope had been found. When the rope is activated in the puzzle room it vanishes (into the inventory of the player) and the variable is set to hold the value of 1. Enter the following Object Condition for your rope object:

```
IF ACTIVATED?
THEN INVIS (18)          (number of the rope)
PRINT (" YOU FIND SOME ROPE",6)
SETVAR (25,V35)          (variable to set the time
                          that the text is held on screen)
SOUND (4)                (a ping noise for effect when
                          you get the rope)
SETVAR (1,V42)            (Setting our variable to tell
```

the computer that the player has the rope!)

ENDIF

You now come to the cliff where the player should ACTIVATE the post to descend the cliff. Enter the following on the Object Condition for the post:

```
IF ACTIVATED?
THEN IF VAR>? (V42,0)    (if you haven't found the rope)
THEN SOUND (3)           (then just make a sound
                          when activated)

IF VAR=? (V42,1)         (if the player has the rope)
THEN PRINT (" YOU ATTACH THE ROPE TO THE POST ... ",5)
```

etc... there now follows all the conditions which take the player down the rope into the dinghy ready to sail to the lighthouse - the rest of the conditions are too long to print here but can be examined if you check out the object condition for the post in the Beachyhead area of the Kitgame.

As you can see, variable 42 has been used in its simplest form just to allow the system to know if the rope has been taken or not and telling it what to do in either situation.....Mandy

**Dear Mandy**

I had the pleasure of phoning you today and, if you remember, asked if it might be possible to issue basic, step by step, instructions of the sort found in handbooks:

1. Switch on the machine.
2. Wait for the following picture to appear.
3. Insert disk A.
4. If you want a pyramid see page 25.  
If you want a sphere see page 30.  
etc, etc.

Personally I am not into games at all but should like to design a house, a car, a "what have you". Allow me please to make a general observation on the handbooks issued with software. The handbooks are clear if you already know how the whole thing is supposed to work but very few people, not brought up with computers, understand them first or even fifth time round. Might it not be easier to begin the handbook with a "hands on", "foolproof", "designed for ignoramuses", flow-chart kinds of instruction leaflet? Once the victim has struggled through the "flow chart", he knows what to expect and can thereafter study the handbook proper. I hope you don't find this suggestion impertinent?

**John Mensink**, Pembroke, Dyfed - AMIGA

I didn't find your suggestion impertinent at all, John. I know exactly what you mean about handbooks. In fact, when we first discussed the manual for the Kit and its layout etc our main concern was that it should be clear and easy to understand. Obviously we didn't succeed in many ways and the restriction in the number of pages allowed for the manual didn't help a lot. However, I do understand exactly what you mean as I too sometimes long for a manual to explain a manual! I hope to provide the sort of information you want in the Newsletters as we cover each different subject of the Kit and to try and cater for the absolute beginner as well as the more experienced user. I'll do my best and hope that the other members will help me in this perhaps by sending in some step-by-step guides to various items under the menu headings, aimed at beginners. It would help a great deal and various subjects can be covered in each Newsletter.....Mandy



**Dear Mandy**

The last time I wrote I mentioned writing again with my problems. Your No 7 Newsletter is to hand for which many thanks, and after seeing your "User Group Game" (Part 1), I feel now is the time. To put you in the picture as to the kind of person you are dealing with, just a few facts. My set-up is an A3000 (Archimedes) with 2MB and double 3.5" drives with Star LC10 printer. Have just started my eighty-fourth year (I know!) - but computers and I are like two peas in a pod, don't laugh, but my first one was a hex. keyboard with an enter key mounted on a small open chassis). We have come a long way since then and I hadn't a clue as to what hex. was! My object in purchasing the 3DCK was not to play and build games. What interests me really is the Virtual Reality bit. Nevertheless I will be building up your game for it has to be a good exercise for me. I do have to confess that I do not spend a lot of time with the Kit (at least, not up till now), perhaps because of the frustration. I do get a bit impatient these days when I cannot get the result I want and can find no documentation to help me. I also have lots of other computer interests like music, painting, simulations, A.I. and expert systems being the main ones. You would think with all that I should be able to cope easily - not necessarily so, even when I have documentation, I often find programmers usually assume too much of the novice. When I saw John Bidmead's letter (A3000) I thought good - until I read it to find he was talking about game and routine testing. However, I may write to him later. But to come to my problems, for I hope I am not wearying you. I can deal with creating the areas, can put something there, like a building (expanded cube), adorn it with door and entrance, open the door and walk through into (and here I was probably a little naive) what I thought was going to be the interior walls of my cube. Imagine my surprise, then, when I found I was out in the field. Now I have tried to build the inside of the cube using some rectangles and sometimes squashed cubes, but to date have had no success. The walls, ceiling and floor look OK from where I am standing but when I move to a different view point I find the back wall, which was my first creation, is far away from the ends of the side walls, and the ceiling is not where I thought it was, sitting nicely on top of my walls, and with a gap one side or overlapping one side. There surely must be a simple way (like all things when you know) of constructing an interior. I do hope you can help me without too much trouble, for I know your time must be precious. I have twenty-four hours every day, little sleep, and still can't find enough of them. Referring to your game directions pp12 Area 1: you mention "You can use the globals to create the walls if you wish." I can find no reference to globals except in index pp61 relating when it cannot be used. There is, as the last item in the pull down "Object" menu a LIST GLOBALS, but the item does not get highlighted nor dealt with in the manual. There is one item that Archimedes users generally are no doubt aware of, but it did puzzle me for a while. It was to do with COLOUR and "getting" with the left mouse button, and "putting" with the right button (see pp8 and pp37/38 of the manual). I was mystified as to why my right button just kept duplicating what the first was doing. We have three buttons on our mouse (the middle one we use for menus) and eventually I found that the programmers had used the menu (second) button for what was described as the right button. So to put a colour or to ACTIVATE use the middle button.

**John Wright**, Baldock, Herts - ARCHIMEDES

I am no spring chicken myself, John. One of the delightful things I have found in dealing with so many people within the world of computing is the wide age range of so many like-minded enthusiasts. I

receive letters from people from around eight years old to people in their nineties. Computers are a great form of bringing all age groups together with a common interest and the only thing I have found that deteriorates as we get older is our dexterity with a joystick and distinct lack of interest in the old arcade "shoot-em-up" type of computer entertainment. When I mentioned that you can use Globals to create your inside walls in the User Group Game I should have pointed out that Globals are only available with 8 bit versions, Atari ST and Amiga versions. The PC and the Archimedes do not have Globals. If an Atari ST or Amiga datafile were loaded into the Archimedes (yes, the Archimedes is able to do that without problem), the LIST GLOBALS would appear on the pull-down menu but would NOT be available to the Archimedes user as the Globals wouldn't actually be there, just the menu item. The only solution would be to create your own Globals - see the article on Globals later on in the Newsletter for further help.....Mandy

**Dear Mandy**

In reply to Margaret Christmas's letter in issue no. 7 about using Graphic Workshop to convert to IFF/LBM files. I have had the same problem with Graphic Workshop 4.6 which was distributed with PC Today Magazine in January 1991. The problem is with Graphic Workshop's compressed IFF/LBM format. There are two solutions to the problem:

1. Set up GWS to generate uncompressed IFF/LBM images - run GWSINSTL and select IFF/LBM compression OFF.
2. Get a newer version of Graphic Workshop. I now use version 6.1 which works fine with both compressed and uncompressed IFF/LBM images.

**Stephen Cross**, Bishop's Stortford - PC

**Dear Mandy**

I feel that I have to write to you to congratulate you on an incredible piece of work. The User Group magazine has been as well put together and presented as the Kit itself. It has been informative and fun to read and, although I myself am not a technical person and have not as much as put two blocks together on one screen, it has filled me with a longing to try. Is it possible for real computer idiots like me, to buy an easy step by step guide to doing just about everything that is available in the Kit. If not, why not? Anyway, congratulations again and keep up the good work.

**Darren Griffiths**, Coventry - AMIGA

I wish that I could write the kind of guide that you want, Darren. Unfortunately I just haven't got the time to devote to such a project at the moment but, when I have finished testing 3DKit II I fully intend to get started on one as so many people seem to want one. I am trying to do the same sort of thing within the Newsletters whenever I can such as with the User Group Game and I hope you will give it a go. Might I suggest that the best way for you to start creating something and to get yourself really into things is to try and recreate a room in your own home - or even your whole house if you are feeling ambitious - perhaps the room where you do your programming? You can then look around and see what needs building and where you are going wrong if anywhere. Start on the design of your room only without any puzzles or conditions, just to get the feel of how everything works then you will soon find yourself wanting to expand your knowledge and start creating something different. To give your confidence a boost just remember that if you enter ANY of the conditions you find in the newsletters you are able to tell everyone that you have done some programming in "C" language!.....Mandy

MORE LETTERS NEXT NEWSLETTER!



## THE USER GROUP GAME!

### PART TWO

Now that we have created our "skeleton" game we are ready to start entering some of the main features. But first, I told you all to enter an IF COLLIDED? or IFHIT condition for the rectangle in the ceiling at the top of the steps in the Dungeon. I have since realised how difficult it is to collide with an object in the roof! So I suggest that you change that particular condition to read: IF ACTIVATED? or for 8 bit, IFACTIVE instead of IF COLLIDED? or IFHIT.

Now we are going to create our problem with the moveable bookcase in the Alchemist's Chamber which is triggered when a particular book is activated by the player. This problem involves animation for 16 and 32 bit users and simulated animation for 8 bit users. First create a bookcase with a couple of shelves which covers the exit to the Solar. First for 16/32 bit users: For the animation we will allow the whole bookcase to glide to one side revealing the doorway so we should make a GROUP. Select CREATE and select GROUP from the object icons. A group will be created with a number and shown on screen so select this. You will be presented with a list of all the objects currently in your area. To make things simpler you might like to RENAME the objects which form part of your bookcase (don't forget the books too!), so they are easy to select from the list do this by selecting ATTRIB icon and clicking on each object of your bookcase in turn and where you see Object Name, click with the mouse and a cursor will appear allowing you to change the name. At the same time make sure you change the STATIC icon on each object to MOVEABLE. Now select each object that forms part of your bookcase by highlighting them by clicking on them with the mouse. When you have selected them all then click on the TICK at the top left of the list and your new group will be created.

For 8 bit users the only method I could come up with to create the same situation would be to have two identical bookcases one, with all the component objects visible, covering the doorway and another identical bookcase with all the objects invisible positioned at the side of the doorway. To create the simulated animation we would enter the following condition:

```
IFACTIVE    o    number of our book object
THEN
INVIS       o
INVIS       o    all the objects in the
INVIS       o    visible bookcase
VIS         o
VIS         o    all the objects in the
VIS         o    invisible bookcase
ENDIF
```

For 16 and 32 bit animation conditions we need to enter the following object condition for our book as follows:

```
IF ACTIVATED?
THEN STARTANIM (1)
ENDIF
```

Create an animator via the top area menu bar and then select edit animation and enter the following condition: Note that in this animation we are going to program it so that when the book is activated the first time the bookcase will slide to one side and if the book is activated a second time the bookcase will close again.

```
ANIMATOR (1)
INCLUDE (g) the number of our group
```

START

LOOP (30) this depends on the width of the bookcase! Check how many steps it takes for the bookcase to clear the door (you can count them if you watch the animation, and adjust as required.

SYNSND (3) Amiga explosion sound

MOVE (10,0,0)

AGAIN

STOPANIM (1) end of first part to move right

INCLUDE (g) same as above

STARTANIM (1)

LOOP (30)

SYNSND (3)

MOVE (-10,0,0) note it's the other direction

AGAIN

STOPANIM (1)

RESTART

Now we are going to turn our attention to the programming for the trapdoor. For this we need a key. In the next newsletter we will deal with separate problems with hiding the key but for now all we need to do is to create a small object in the Solar to represent our key. In my case it was object 6. We need a condition to represent the player taking the key so enter the following for our "key" object:

16/32 BIT	8 BIT
IF ACTIVATED?	IFACTIVE 6
THEN INVIS (6)	THEN
ENDIF	INVIS 6
	ENDIF

Go to the Great Hall and create two flattened cubes, one to neatly cover the open trapdoor (black rectangle) and one flipped to stand upright beside it as if it were an open trapdoor. Make the second invisible as this will only become visible when the trapdoor is "opened". For the following conditions let us say that the closed trapdoor is object 8 and the open trapdoor is object 9 and enter the following conditions for them:

```
16/32 BIT
IF ACTIVATED?      if player activates the closed trapdoor
THEN IF INVIS? (6,5) and if the key in the solar is taken
THEN INVIS (8)      make closed door invisible
VIS (9)             make open door visible
SYNSND (5)          Amiga thud sound
ELSE SOUND (4)      or if the key isn't taken make ping sound
ENDIF
ENDIF
```

```
8BIT
IFVIS      6      5      this is to check if the key has been taken
THEN
ELSE
END
ENDIF
IFACTIVE  8      if it has been taken then
THEN
INVIS     8      if the closed trapdoor is activated
VIS       9      then
ENDIF           make it invisible
                and make the open trapdoor visible
```

Next issue we will add more puzzles so watch this space!

## STEP-BY-STEP GUIDE

### MAKING INTRODUCTORY/LOADING SCREENS AND MUSIC ON THE AMIGA!

By *Mieke Van Der Poel* - AMIGA

Firstly, on my PD disk of MOUNTAIN ADVENTURE you can find all the files and the start-up sequences that are needed for loading screens, music etc. I followed the instructions of Paul Ramsell published on pages 21/22 of Newsletter 3 but I used Protracker and Noiseplayer instead of *ed* and *Medplayer*. I also bought the PD disk by Andrew Herbert (E.F.T.P.O.T.C.M.) which contained all the utilities I needed such as *PICSHOW*, *PICEND*, *THAR*, *NOISEPLAYER*, *POWERPACKER* LIBRARY and his start-up sequence. The only difference in my case is that each time you boot the disk you have to go through the loading screens, whereas on Andrew's disk he gives you the choice: F1 to play the game, F2 to get instructions. I couldn't figure out how he programmed that, so perhaps he can write about it sometime? (Come on, Andrew, we wait with baited breath! - Ed). Well, here is how to go about it:

Firstly make a nice picture that is relevant to your game. It does not matter in which size or with how many colours you make it. You can even make a picture in 3D-Kit, save it as an .IFF file by pressing F2 and giving it a name. You can then use it as it is or load it into your art package and make the changes or adjustments that you want and save it again. In the art package you can choose a suitable font to make a text file containing the title, the author, etc.

The next picture could be an instruction screen on how to play the game. You could do these text files in 3D-Kit also using the text animation and condition that is included in the Kitgame and make a text instrument for it.

As for sound effects: Most music packages do have a couple of modules included from which you can derive sound samples and save them separately on a disk, so you can include them into the *MakeSampleBnk* program. If you are not that good in making a music module for your game you can choose from several PD Libraries which have special music disks containing modules and even sound samples (see *Amiga Format* or *CU Amiga* etc).

So what you need is:

1. Your runnable game including the sound sample bank.
2. Several pictures/text pictures.
3. A music module.
4. PD utilities such as: *picshow*, *picend*, *noiseplayer*, *powerpacker* library, *TBar* and the commands *CLS* and *WAIT*.
5. A start-up sequence.

NB: Before using *Noiseplayer*, try it out once by typing *Noiseplayer* (Enter) and you will be presented with a little explanation on how to use the program. These are, for instance, different numbers for keys to be pressed when you want to stop the music.

Somewhere in one of your introduction screens you must remind the player to stop the music by pressing a key before the game is loading because otherwise he ends up with a GURU!

Work as follows:

(When you have a disk utility such as *Diskmaster* all the work hereafter is much easier!)

LOAD in your workbench and format a blank disk and change the name *EMPTY* into a name you want, for instance the name of your game.

When you have just one drive you can copy some commands on the ram disk in the Amiga's memory using the CLI. So open your workbench window and go into *SHELL*. Enter the following lines:

```
cd c                (Enter)
copy copy ram:      "
copy dir ram:       "
copy mkdir ram:     "
copy install ram:   "
copy delete ram:    "
copy cd ram:        "
copy list ram:      "
copy ed ram:        "
```

```
Put your blank disk in df0 and type: cd df0:      (Enter)
Delete Trashcan files:                          delete Trashcan "
                                                    delete Trashcan.info "
```

If you use *Workbench 1.3* you must include a disk validator file in a directory *L*, so you can check this out by typing: *dir* (Enter) and if it is there type: *cd l* (Enter), type: *list* (Enter) and the file *Disk-Validator* appears on your screen. If it is not there type: *mkdir l* (Enter).

First we make the other directories we need and then we go on with copying the necessary files:

```
mkdir c             (Enter)
mkdir Libs          "
mkdir S              "
```

```
Type: cd ram:      "
Type: list         "    and you will see the files you copied on the
                        ramdisk.
Type: install df0: "    You have made your disk bootable now.
Type: copy ed df0:s "    You have the text editor for your start-up
                        sequence in the right directory.
```

Now in order to have enough memory to copy the necessary files to the ramdisk we delete the files we do not need anymore:

```
Type: delete mkdir (Enter)
Type: delete ed    "
Type: delete install "
```

Change your blank disk with the disk your runnable game is on in drive *df0*. Copy all the files of the game to ram: in the same way as described above. Then swap the game disk with the blank on and type: *cd ram*: (Enter).

Then copy all the game files to *df0*:. Delete them on the ram disk.



You do the same with the music module, the picture files and the file Noiseplayer. Having done that and you are still in the ramdisk mode, and you do not have the Disk-Validator on your game-to-be disk you put in the Workbench disk, type: cd df0: (Enter), open the Workbench window, go into the SHELL and type:

cd L (Enter)

Copy the Disk-Validator to ram: copy Disk-Validator ram: (Enter)

Then put the disk with the utilities in drive df0 and copy to ram:

```
copy cls ram:          (Enter)
copy picshow ram:      "
copy picend ram:       "
copy wait ram:         "
copy Tbar ram:         "
copy Powerpacker Library ram: "
```

Put your game-to-be disk in df0 and type: cd ram: (Enter)

```
type: copy cls df0:c      "
type: copy picshow df0:c  "
type: copy picend df0:c   "
type: copy wait df0:c     "
type: copy Tbar df0:c     "
type: copy Powerpacker Library df0:libs "
Event. type: copy Disk-Validator df0:L "
```

Now we are going to make the start-up sequence by using the text editor ED. So type:

```
cd df0: (Enter) cd s (Enter). Then you type:
ED startup-sequence (Enter)
```

Now type in the following:

```
cls
picshow pic1 (or your picture name)
noiseplayer -p2 mod.name (name of the module)
picshow pic2 (your own picname)
picend pic1
wait 20 (a number of seconds you want pic2 to appear)
picshow pic3 (your own picname)
picend pic2
wait 30
picend pic3
wait 01
cls
Tbar Please wait LOADING
Gamename (the name of your game)
```

Then press ESC X and again ESC X to save and leave the text file.

It is now in the right directory and you can delete the file ED if you wish. Now if all goes well your disk will boot up with the introductory screens, the music and the game.

Next issue Mieke shows us a step-by-step guide to doing exactly the same on the PC. So watch this space!!.....Mandy

## DOWN TO BASICS

By Frank Carr - ATARI ST

I've been using BASIC for five years in the form of Fast Basic, STOS and now 3DCK BASIC and, believe me, 3DCK is the easiest. To answer the question what to do after playing the Kitgame, the newcomer really needs to understand why Incentive put it there. Load the Kitgame into the Editor, from there you can access every part of the program, you can check General Conditions, Area Conditions, Attributes for each object and Object Conditions. I would recommend that newcomers take this line, for example, from the Kitgame in Area 1 go to the door that leads to the puzzle room - the one with the moving bricks behind it. If you activated the door it opens, no go to conditions, click on the door and up comes the program for that object. Nearly all the Object Conditions require an IF, THEN and ENDIF loop. Basically it means IF that block is shot THEN invis that block ENDIF. If the object is activated then start animation.

If you break down most programs in 3DCK that's all it is within reason. The biggest problem seems to be the word VARIABLES. For members who understand programming it is simple but for the novice it is a nightmare! It is difficult to explain on paper but I will try. We need a small program to demonstrate it:

### TUTORIAL STEP BY STEP

We need a room with a table in the middle and a door at one end plus a key on the table. Create a cube, edit cube, stretch cube 1000, make 400 high and shrink to 20. The screen dialogue should read S:1000,0400,0020. Now move it down to ground level, copy cuboid 2 above, edit it by turning sideways and moving into position, repeat for the other two walls, colour the walls so they stand out. To create a table, shrink a cuboid to 200,20,200. Create a rectangle for the door coloured black. Create another rectangle for the key (shrunk to 20,20) and place it on the table. Stand in the room, say, at the end of the table, facing the door and create an entrance which will be entrance two.

Load KGBorder and go to General Condition and click on SET VIEW WINDOW. Set the window to the size you want within the border screen. Click on controls, click on Move Forward with the LEFT mouse button, click with RIGHT mouse button. You will be presented with the border screen again, use left mouse button on the top left corner of the forward arrow icon and drag to the opposite bottom corner and click left mouse button. Select EITHER from the menu. Repeat the procedure down to Rotate Right.

Now we will set up a Text window at the bottom of the screen. Create Instrument and Edit Instrument (1). Click on Type until it comes up TEXT WINDOW. At X POS enter 060, at Y POS enter 189. At LENGTH enter 244, height enter 008. Click on View Box to show you where it is positioned. One other thing, it is a good idea to go to ATTRIBUTES and put a name to your objects, table, door, key etc.

Now we are ready to program:

```
Condition for DOOR:
IF VAR=? (V40,2)
THEN IF COLLIDED?
THEN GOTO (1,1)
END
```

```
ENDIF
ENDIF
```

```
IF VAR=? (V40,1)
THEN PRINT ("          YOU UNLOCK THE DOOR          ",1)
DELAY (100)
PRINT ("          ",1)
SETVAR (2,V40)
END
ENDIF
```

```
IF ACTIVATED?
THEN PRINT ("          DOOR LOCKED          ",1)
DELAY (100)
PRINT ("          ",1)
ENDIF
```

Condition for KEY:

```
IF ACTIVATED?
THEN PRINT ("          YOU HAVE THE KEY          ",1)
SETVAR (1,V40)
DELAY (100)
INVIS (8)
PRINT ("          ",1)
ENDIF
```

Let us examine the program, three routines for the door and one for the key. Look at the door routine first: When activated it ignores the V40 routine because we haven't called it yet and it uses the locked door routine. Now we need the key. On activating the key we set variable 40 to hold the value of 1. We then go back to the door and it now uses the unlock door routine and sets variable 40 to hold the value of 2, which then uses the first routine to tell it to go to Area 1, entrance 1. Notice that the routine has two ENDIFs. If there are two IFs then there should be two ENDIFs. Save it and run it. To re-run it just hit ESCAPE key as this sets all variables to zero. Press F1, select Area Menu and select GOTO ENTRANCE. Click on 002 and select TEST.

We have used variable 40 as our counter. At 0 the door is locked, we set it at 1 when we picked up the key. We set it to 2 when we unlock the door so that we can go through the door to another area.

Variables are just counters. Study the program, play around with it, alter the delays, take out the END to see what happens. Try to understand the program.

Here is a tip for people building up the User Group Game: In the Area Conditions enter:

```
IF VAR=? (V16,0)
THEN PRINT ("          ENTRANCE HALL          ",1)
SETVAR (0,V16)
ENDIF
```

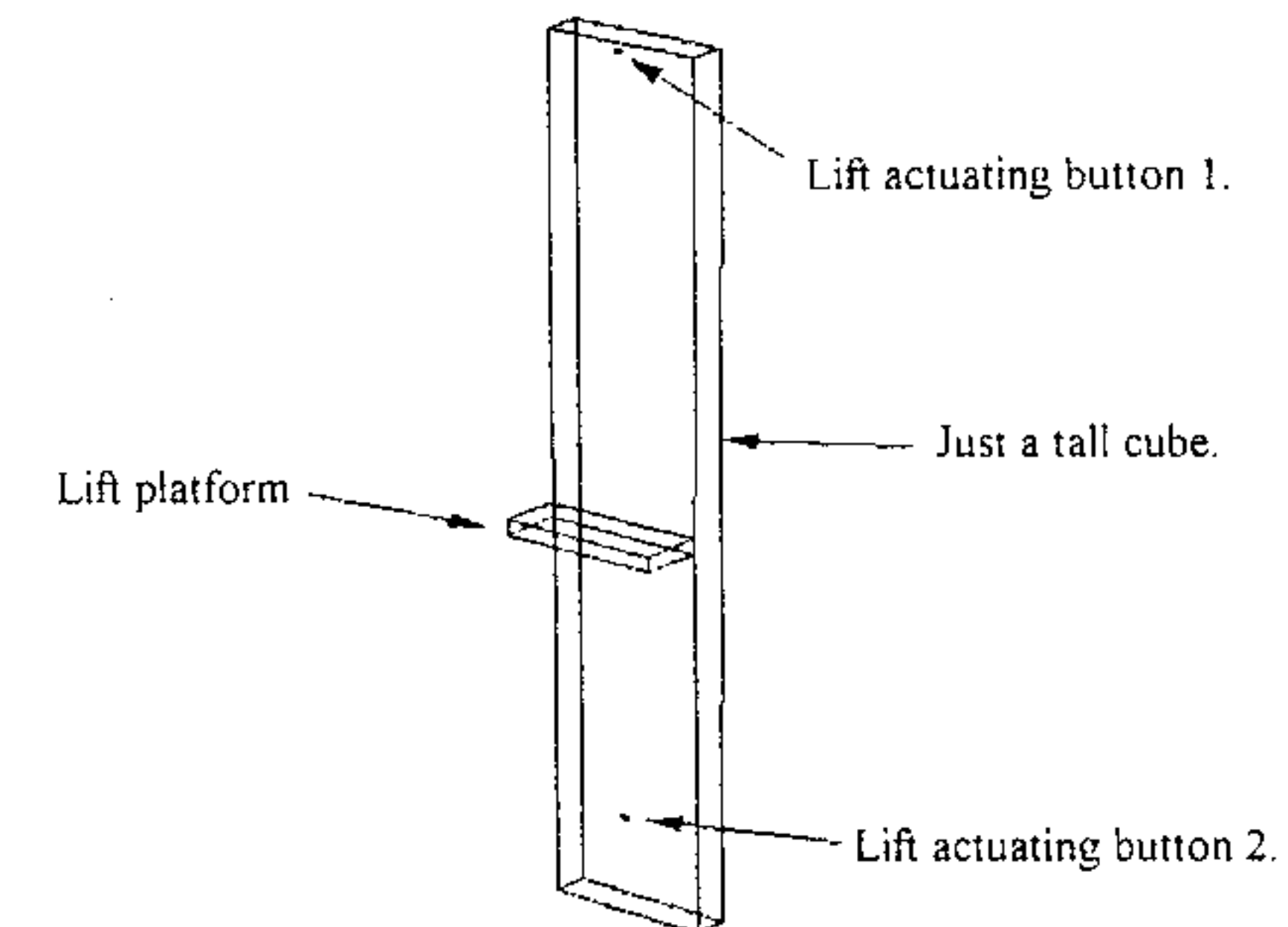
Do the same for all the other rooms (areas).

(Thanks, Frank, for this article. If any other readers would like to write a similar article to explain in a step-by-step manner any other feature of the Kit to help fellow members it will be greatly appreciated, I am sure, especially for beginners.....Mandy)

## 16/32 BIT ROUTINES

### GIVE YOURSELF A LIFT!

By Mark Rose - PC



#### Conditions and Animator

- |   |   |
|---|---|
| 1. Area condition                       | - STARTANIM(1)  |
| 2. Ground condition                     | - IF COLLIDED?<br>THEN SETVAR (0,V100)<br>ENDIF   |
| 3. Lift button condition (both buttons) | - IF ACTIVATED?<br>THEN TRIGANIM(1)<br>ENDIF  |
| 4. Lift platform condition              | - IF COLLIDED?<br>THEN SETVAR (1,V100)<br>ENDIF   |
| 5. Upper floor condition                | - IF COLLIDED?<br>THEN SETVAR (0,V100)<br>ENDIF   |
| 6. Animation (1)                        | - INCLUDE (lift platform object)<br>START<br>WAITTRIG<br>WAITTRIG<br>LOOP (247)<br>IF VAR=? (1,V100)<br>THEN ADDVAR (30,V1)<br>ENDIF<br>MOVE (0,30,0)<br>AGAIN<br>SOUND (5)<br>WAITTRIG<br>WAITTRIG<br>LOOP(247)<br>MOVE (0,-30,0)<br>AGAIN<br>SOUND (5)<br>RESTART |



The variable for the users verticle location is only changed if the user is actually standing on the lift platform. So the lift can go up and down without you needing to stand on it.

You can also go up the lift, step onto the upper floor platform (not shown on the diagram) and press the button for the lift to descend without you on it. Of course you have to press the button again for the lift to come back up if you want to get down.

Just enter the conditions as I have written them and don't forget the ground condition. This is actually on Cuboid 1 because if you press the lower button and the lift ascends, you will not be able to move without the condition on the ground.

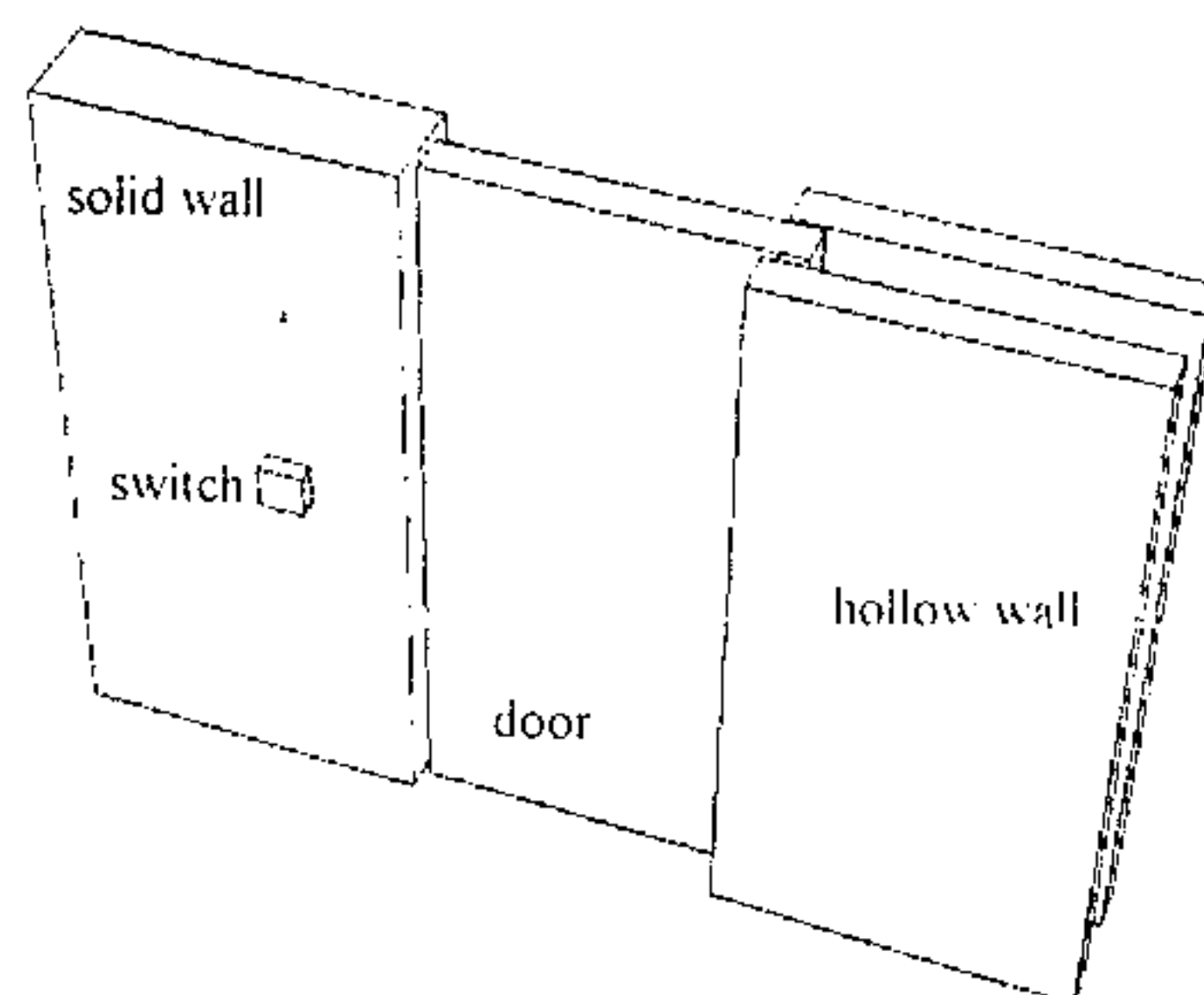
Again the condition on any higher level must be included (the same as the ground condition), or again, when the lift platform is moving you will not be able to move. This variable used here just sets the "is user standing on the platform" variable to true or false. If true, then the animator includes the ADDVAR to the users verticle value. Hence you then move up with the lift platform.

The loop value of 247 is just the number of times the sequence of commands is executed and can be changed to suit the individuals needs. You can also change the amount of movement. This will either speed up or slow down the movement of the lift platform, but it is important to make sure that the amount that you add to V1 (users verticle position), is the same, or strange things happen.

I am still working on improving this routine and future upgrades will include the ability of the lift platform to stop at various floors on the way up or down, depending on the users floor number selection.

### A VARIATION!

By Mark Rose - PC



```
Area Condition -      STARTANIM (1)

Switch Condition -    IF ACTIVATED?
                      THEN TRIGANIM (1)
                      ENDIF

Animation (1) -        INCLUDE (sliding door object)
                      WAITTRIG
                      WAITTRIG
```

```
SOUND (8)
LOOP (30)
MOVE (10,0,0)
AGAIN
WAITTRIG
WAITTRIG
SOUND (8)
LOOP (30)
MOVE (10,0,0)
AGAIN
RESTART
```

As you can see, this is just a variation of the lift routine. The loop value is dependant on the distance between the walls. The sound (8) is from the PC version of 3DKit and it sounds for the duration of the animation. A second switch with the same condition as above is (obviously) necessary on the other side of the solid wall. Of course it is possible to add other objects to the door and create a door similar to that found in films like StarTrek for example.

Next issue will feature more 16/32 Bit routines including a feature telling you everything you need to know about messages by J. Hayes so watch this space!.....Mandy

\*\*\*\*\*

### HALL OF FAME

\*\*\*\*\*

My sincere thanks to all the following members who took the time and trouble to send in contributions for the Newsletters during the past two months:

STUART SHELTON, STEPHEN CROSS, TOM REILLY, DANIEL FOSTER, DARREN GRIFFITHS, FRANK CARR, PETER LAWS, M. HARRIS, MARK ROSE, J. HAYES, DAVID SAMBROOK, MIEKE VAN DER POLL, STEVEN FLANAGAN, HUMPHREY BOLTON, ROBIN BALL and STEPHEN FORSYTH.

ALL members are cordially invited to send in contributions for the Newsletters. If you have just a hint or tip, routines of any kind, complex or simple, or a full-blown article they are all most welcome. Don't worry if your routine doesn't appear in the immediate issue, it will appear in the following one. ALL contributions will be printed, nothing is ignored! Let's hear from you! There are hundreds of members and most of you appear to be quite shy - don't be, we would love to hear from you so do write in. Advanced users are waiting for more complex routines to give them ideas and the beginners are all waiting for step-by step routines and articles to get them off the starting line. It would be generous to share your prized routines, hints and tips with fellow members so, everyone, try to put pen to paper (or fingers to the keys) in time for the next deadline which is the 1st October 1992 ready for the October/November issue.

\*\*\*\*\*

## 8 BIT ROUTINES

### TRANSFER VARIABLE VALUES By *Steven Flanagan* - COMMODORE 64

Many times I have wished there was a TRANSV command in the FCL to transfer the value from one variable to another but there isn't. When I was writing an experimental routine I discovered a way. It is so simple, I can't believe that I haven't thought of it before, or that none of you have written in about it. The routine below transfers the value of variable 1 to variable 2.

```

SETV 0 2      TESTV 16 1
TESTV 1 1      IFEQ
IFEQ          THEN
THEN          ORV 16 2
ORV          ENDIF
ENDIF        TESTV 32 1
TESTV 2 1      IFEQ
IFEQ          THEN
THEN          ORV 32 1
ORV          ENDIF
ENDIF        TESTV 64 1
TESTV 4 1      IFEQ
IFEQ          THEN
THEN          ORV 64 2
ORV          ENDIF
ENDIF        TESTV 128 1
TESTV 8 1      IFEQ
IFEQ          THEN
THEN          ORV 128 2
ORV          ENDIF
ENDIF        XOR 255 2
ENDIF        END

```

What the routine does is to test each bit of variable 1 and if the bit is set then it will set the same bit in variable 2. I will not attempt to explain binary here as it is very well documented in issue 3 by George Dixon. TESTV tests a bit in the variable and returns a truth if the bit is 0 (clear). For example:

```

TESTV 1 1      - tests bit 0 of variable 1.
IFEQ
THEN
..... any commands here will be executed if the bit was
          0 (clear).
ELSE
..... any commands here will be executed if the bit was
          1 (set).
ENDIF
..... any commands here will be executed regardless of
          the state of the bit.

```

The routine can be in a General, Local or Procedure routine depending on its use. You can change the variables used in the TESTV and ORV commands to any variables you wish. This routine is a lot more useful than you might think. It opens up many new possibilities for the Kit. For example the Information Screen that has been talked about before was just about impossible to program until now. All you have to do is transfer the viewpoint variables and area number into storage

variables, goto the area with the information screen and when you want to return to the program, transfer the stored values back into their variables to begin from exactly where you left off. It allows you to store results of tests for future use.

I hope you put this routine to good use, and if you do use it, please write in to Mandy to let her print it in the Newsletter so everybody will learn about it.

(In the next Newsletter Steven gives a complex and most useful routine to compare the values of variables so watch this space!.....Mandy)

### **CHANGING THE FONT** By *Stuart Shelton* - SPECTRUM

In this program, by altering variables 126 and 127 you can have fully italic text at all times. First you should enter this BASIC program:

```

10 PAPER 0: BORDER 0: INK 0: CLS
20 CLEAR 64599: LOAD "chars," code 64600,768
30 POKE 23606,88: POKE 23607,251
40 LOAD""

```

SAVE "3D LOADER" LINE 10

Then save it at the start of the tape. Reset then type all of this! There are two ways of doing this next bit, I'll show here the easiest one to type. The other way is change lines 50.90 and 130 of the following to "READ B" instead of "INPUT B" and add (linenumber) "DATA" (numbers), the advantage being that you can see if you have made a typing error.

```

10 FOR A=15616 TO 16383
20 POKE 48984+A,PEEK A
30 NEXT A
40 FOR A=1 TO 80
50 INPUT B (READ B)
60 POKE 64727+A,B
70 NEXT A
80 FOR A=1 TO 208
90 INPUT B (READ B)
100 POKE 64683+A,B
110 NEXT A
120 FOR A=1 TO 208
130 INPUT B (READ B)
140 POKE 65119+A,B
150 NEXT A

```

Then if you are using the READ B version, continue with

```
160 DATA 0, 60, 70, 74, 148, 164, 120, 0
```

When you are ready type RUN and the screen will either go blank for a couple of seconds and then say 0 OK. 150:1. If you are using the other version the screen will immediately have a flashing C at the bottom requesting you to type in the next number. Type all of the data (it acts like the usual redefine graphics data when you are programming, so if you used this you could have 97 user defined graphics, just change the numbers. When you have typed all the data in save it using:



SAVE "chars" CODE 64600,768  
 Save it next on the tape after the loader. Reset your machine and load your 3D Kit game's DATAFILE. Make your first area all black and set all of your variables in this area before going to area 2. The condition should look like this:

LOCAL CONDITION 1:  
 (set all vars.)

```
SETV 008 126
SETV 251 127
CROSS 0
GOTO 1 2
END
```

Then IMMEDIATELY save it. Load in the compiler and save your finished game on the tape after the character code. Reset and just load it in and watch it go! Note: Variable 126 acts as address 23606 and variable 127 acts as address 23607. The normal setting for addr 23606 is 0 and 23607 is 60. By changing these two variables you can chop and mix text.

#### LISTING 1 (DATA FOR NUMBERS)

```
0 60 70 74 148 164 120 0
0 48 80 16 32 32 248 0
0 28 34 4 56 64 124 0
0 30 4 24 4 72 56 0
0 6 10 20 36 126 8 0
0 30 16 60 2 68 56 0
0 30 32 60 66 68 56 0
0 30 2 4 8 16 32 0
0 28 36 56 68 68 56 0
0 28 34 34 28 4 56 0
```

#### LISTING 2 (DATA FOR CAPITALS)

```
0 12 18 34 62 66 66 0
0 28 18 60 34 66 124 0
0 28 34 32 64 68 56 0
0 24 20 34 34 68 120 0
0 30 16 60 32 64 120 0
0 30 16 60 32 64 64 0
0 28 34 32 76 68 56 0
0 18 18 60 36 72 72 0
0 62 8 16 16 32 248 0
0 2 2 4 68 72 56 0
0 18 20 55 40 68 66 0
0 16 16 32 32 64 124 0
0 34 54 42 66 68 68 0
0 18 26 42 44 68 68 0
0 28 34 34 68 68 56 0
0 28 18 34 60 64 64 0
0 60 66 66 164 148 120 0
0 28 18 34 60 68 66 0
0 28 34 24 4 68 56 0
0 62 8 8 16 16 32 0
0 17 34 34 68 68 56 0
0 34 34 36 36 40 16 0
0 33 33 66 66 90 36 0
0 34 20 24 56 68 130 0
0 34 20 8 16 32 64 0
0 62 4 8 16 32 124 0
```

#### LISTING 3 (DATA FOR LOWER CASE)

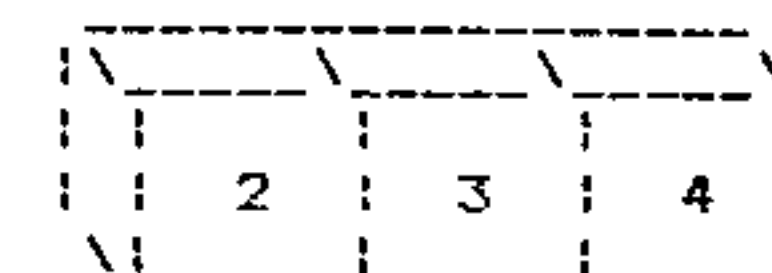
```
0 0 12 2 60 68 56 0
0 16 16 60 34 66 124 0
0 0 28 32 32 64 56 0
0 2 2 28 36 68 56 0
0 0 28 34 124 64 56 0
0 6 8 12 16 16 32 0
0 0 14 18 34 60 4 120
0 16 16 62 34 68 68 0
0 4 0 24 8 16 120 0
0 2 0 4 4 8 72 48
0 16 20 56 48 72 68 0
0 8 16 16 32 32 24 0
0 0 54 73 73 146 145 0
0 0 60 34 34 68 68 0
0 0 28 34 36 68 56 0
0 0 28 18 34 60 64 64
0 0 30 18 36 60 8 30
0 0 14 16 16 32 32 0
0 0 30 32 24 4 120 0
0 4 30 8 16 18 12 0
0 0 34 34 68 68 56 0
0 0 34 34 36 40 16 0
0 0 65 65 146 146 108 0
0 0 34 20 24 40 68 0
0 0 18 36 60 8 16 96
0 0 60 8 16 32 120 0
```

## KILLER CUBE II - THE REVENGE!

By M Harris - 8 BIT

This routine has the advantage over the original as the player can move between frames and switching is controlled by the same part for f/b movement. The delay of 30 between each switch is just to slow the routine down so you can see what's happening to variable 1 by watching the score counter - useful for debugging routines. On setup block 2 should be visible and all others invisible:

30 OUT 60



```
120 BACK 90
TESTV 128 050 - Check if cube destroyed. If so then go to next
IFLT Local condition. * If you need more cubes make
THEN sure to include these 4 lines at the top of any
END other locals you use to speed up processing.*
ELSE
ADDV 001 001 - Increase frame count.
IFSHOT 002 - If any of the cubes are shot then destroy the
OR lot.
IFSHOT 003
OR
IFSHOT 004
THEN
DESTROY 002
DESTROY 003
DESTROY 004
ORV 128 050 - Switch all anim routines attached to this off.
END
ELSE
CMPV 030 001 - If the frame count is 30, switch blocks 2+3 for
IFEQ forward movement.
OR
CMPV 120 001 - If the frame count is 120, switch blocks 3+2 for
IFEQ backward movement.
THEN
TOGVIS 002
TOGVIS 003
CMPV 120 001 - If full circle then reset and repeat.
IFEQ
THEN
SETV 000 001
END
ELSE
CMPV 060 001 - If the frame count is 60, switch blocks 3+4 for
IFEQ forward movement.
OR
CMPV 090 001 - If the frame count is 90, switch blocks 4+3 for
IFEQ backward movement.
THEN
TOGVIS 003
TOGVIS 004
END
ELSE
IFCRUSH - If player under block then S-P-L-A-T!!
THEN
ENDGAME - This should be replaced with a call to your
END endgame routine.
```

## GLOBALS

On some versions of the Kit, ie. Amiga, Atari ST, Spectrum, Commodore and Amstrad, the programmers kindly provided a special area in which they had already created four walls a roof and a floor to represent the interior of a room. These were placed there so that you could quickly create a room by selecting the Globals and either clicking with the mouse button on which facets you wanted to select or, as with the 8 bit versions, by selecting with a plus or minus sign. Then when you selected OK and providing you had first created a new blank area the Globals you had selected would appear within the new area ready for you to colour and work with in whatever way you wished. On the 16 bit versions this is as far as you could go with the Globals but on the 8 bit versions you have the added ability to create extra Global objects, such as trees etc, to add to the Global area and to call up when you want them.

There are, however, a few drawbacks to using the Global objects in that they are, admittedly, far from perfect and certain problems and error messages can arise when doing so. If you create new Global objects with the 8 bit versions you must remember that whenever you wish to use them they will appear in EXACTLY the same spot on the new area as they were in the Global area which, if you are not careful could make the objects try to appear within the space of another object with dire results. On the Atari and Amiga versions if you select the Globals and use them in a number of different areas and try to colour them differently you will recolour the previous Globals you used in earlier areas. Other little niggles such as the floor changing colour as you walk across it can also occur. I hate to say it but my advice is - unless you really know exactly what you are doing then it is best to avoid using the Globals.

I find that it is much easier for me to create my own Globals on 16 bit - something which I hadn't thought of doing until I received a tip about this from Daniel Foster (PC) who gave me the idea (thanks Daniel!). This can be done quite easily and will enable ALL Kit users to have Globals. Firstly, you should create a blank area to work on and with the aid of expanded and aligned cubes carefully create four walls, ceiling and floor to the size that you think would be most useful. Then create a group and make sure that all the objects APART FROM THE GROUND CUBE (001) are included. Now save the group as an object to disk so that they can be loaded in and coloured as you wish whenever you wish by creating a new blank area and selecting LOAD OBJECT.

When creating your Globals you may find, as some people have, that it is a bit fiddly getting your walls aligned so there are no gaps that the player can walk through. I do this by creating the four walls, ceiling and floor and, before closing them up together I make sure that when I am in ENTRANCE 001 I am in exactly the centre of them. I then draw up the east and west walls to the position I want and then make the north and south walls quite a bit wider than necessary before closing them up tightly against the other walls. I do the same for the ceiling and the floor also. This usually ensures a snug fit in all the corners where the Globals join together, and, if the player is going to be completely enclosed by them they can't see if any are wider than the others anyway!

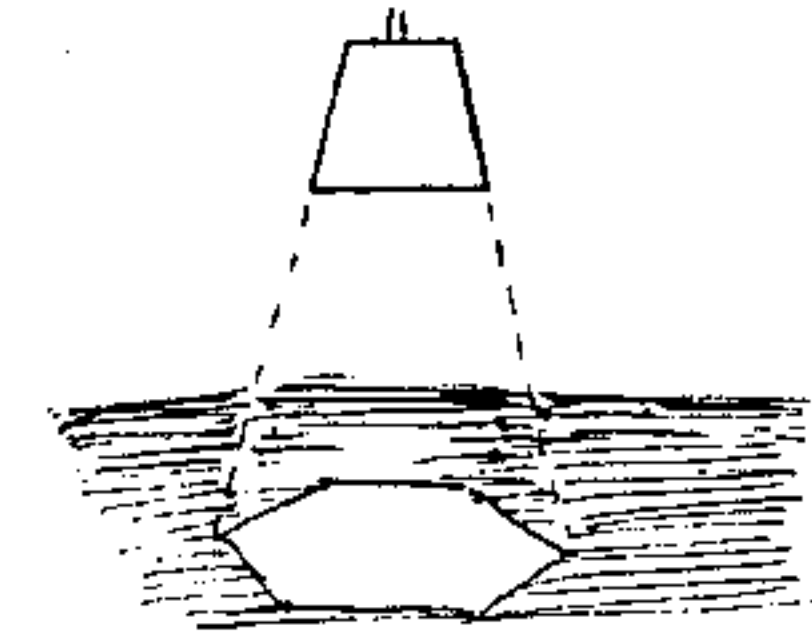
If anyone else would like to write something about Globals, or any other aspect of the Kit then please feel free to send them in....Mandy

## LIGHTING

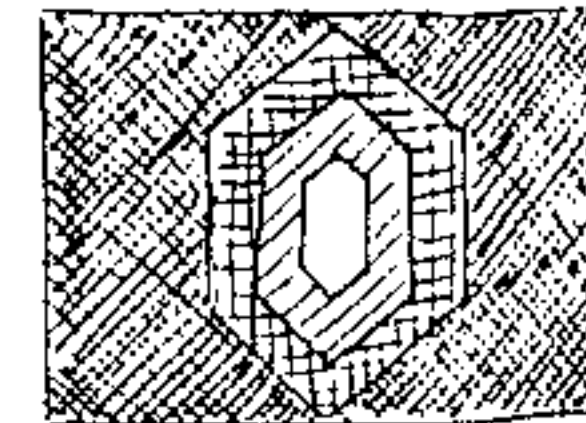
By Robin Ball - ATARI ST

You can improve the look of your environments a lot by adding some lighting effects. The Kit won't do this for you as it takes an incredibly long time for the computer to work out the calculations for light, but you can create the illusion of light very easily if you understand how light works and how it is cast by lamps and such things. It is quite complicated to understand and also to explain. (Thankfully Robin has send in another PD disk with all the lighting effects done for you - I'll review it next issue...ED).

Firstly I will explain how to create a pool of light from a hanging lamp. To start with, create a lamp shade, a single pyramid is ideal, then create a shape for the pool of light, a hexagon will do, it doesn't have to be round. Then place the hexagon on the floor and colour it a shade lighter than the surrounding floor. You must then increase the size of the hexagon. To work out the size it needs to be, simply project the edges of the lamp shade down onto the floor then extend the hexagon to fill that space. This diagram should help.

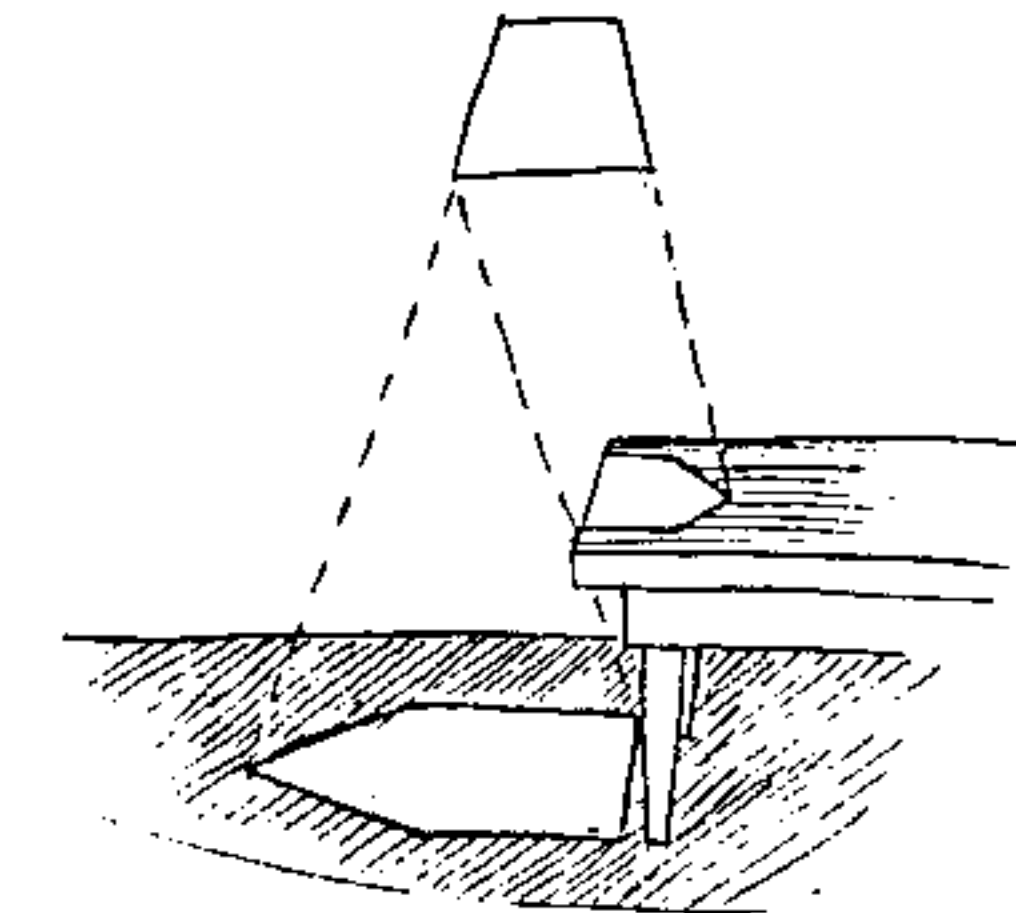


If you haven't used many objects

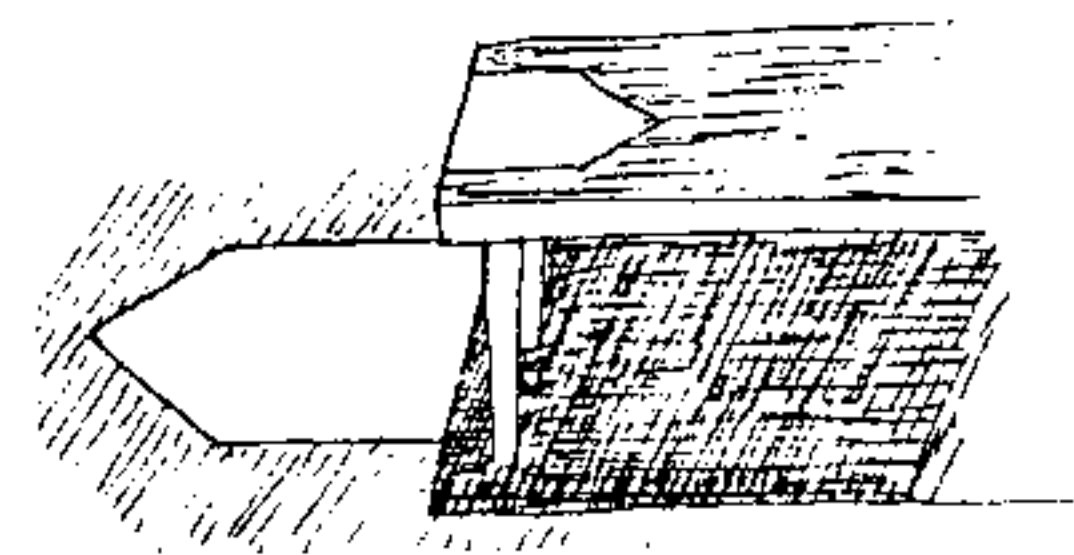


in your area, you can increase the effect by creating several circles inside each other then colour the inside one brightest and the outside one quite near to the floor colour. It should look something like this diagram shown here on the left.

If there are objects in your area, you will have to cast your light over them also, again projecting the rays of light down from their source. You could cast light over the edge of a table for example as I have done in the diagram shown here on the right.

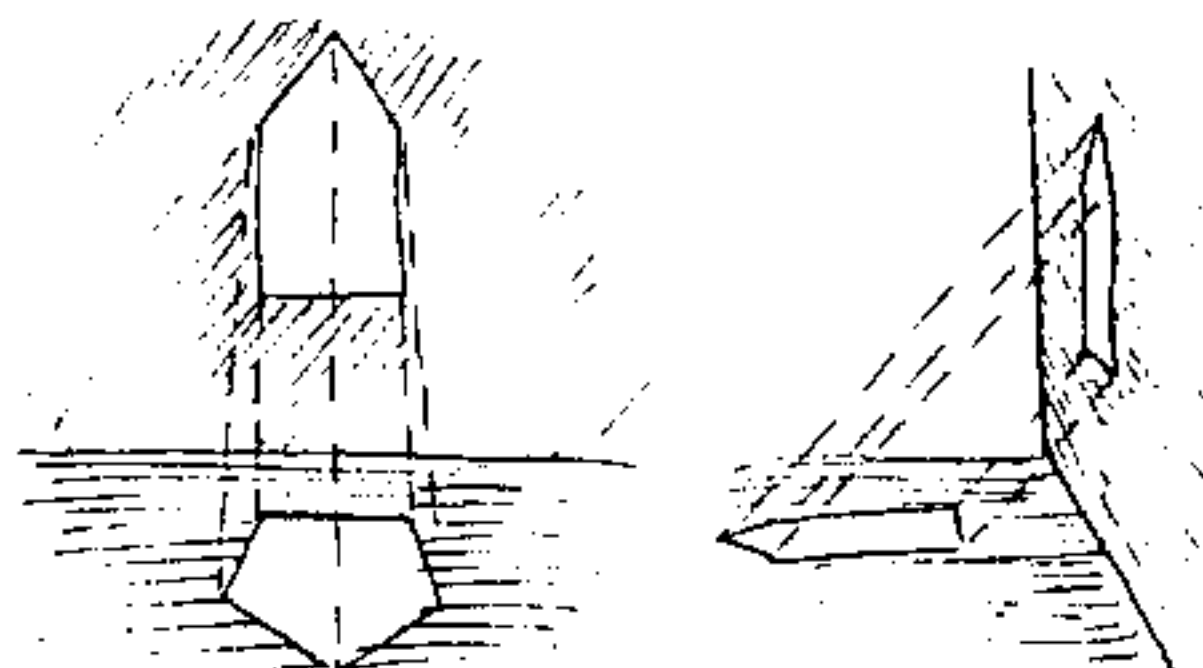


Once you do this you should create shadows as well. It now becomes quite complicated but so long as you remember to project the rays of light from the source over the objects and onto the floor you can work out a pretty good approximation. To help you work out how light falls you could always set up a lamp in a darkened room and place objects in the path of the light.





When you have worked out how light falls from a lamp, you can work out how light from the sun works also. You don't need pools of light of course but some cast shadows would be good. What you could also do is to create pools of light through windows. This is done in the same way as cast light over an object. Simply guess where the light is coming from and create a rectangle on the floor or even from an open doorway. To demonstrate all of these effects I've created a small collection of areas based on the Club Game. It is on the PD disk under the name CLUBGAME1. Practice the light effects, it may be difficult to start with but once you've got the hang of it your games will look much more professional.



\*\*\*\*\*

## HINTS AND TIPS

TIPS by *Andrew Hull* - AMIGA

Write down your variables and what they are used for. This is extremely important when programming in a language such as FCL because your variables have fixed names. A "life counter" for enemies/doors/etc can be implemented by having lots of rectangles stacked behind each other. As each one is shot it disappears revealing a darker rectangle. When the last (black) rectangle is shot then the enemy/object/whatever is de-activated. This saves having to use variables for each enemy. Try to work around the area limitations of the Kit by enclosing your areas to give the player less of a feeling of the game being made from floor tiles! For example, avoid large open areas if you can by enclosing them with barriers, buildings etc. This can really improve the atmosphere of a game. Lighting effects can be created by having two sets of objects, one light and one dark. When the "switch" is activated, swap from light to dark (or vice versa) objects. This can be combined with a timer to re-create day and night.

TIPS by *Robin Ball* - ATARI ST

Firstly a tip for relatively easy but impressive borders. To start with, this will only work with Deluxe Paint ST so get a copy of that (you can get it with Golden Image Brush Mouse for only £25). Create a range of 8 greys, or another colour if you want, from black through to white with black as the background. Then select the third lightest grey and select the medium sized round brush and spray lots of light grey in a thick (about 1/3 screen), strip right across the screen. Make sure that there is more white than black showing through. This will form the basic display panel. It doesn't look very impressive yet but now comes the fun part. Select TO BRUSH from the ellipse menu then draw a circle of about 1 inch across. Select SMEAR from the FX menu and smear the spots together to form an uneven mass. Then when they're to your liking, select SMOOTH from the FX menu and slowly draw over the whole panel. It creates a range of tones between the light grey and the black making a pretty good marble effect. Flatten the

top with a large black rectangle then lighten a two pixel wide line across the top and down the left side by two tones (which is why we used the third lightest grey and not the white). You will have to do this in Zoom pixel by pixel and will take quite a long time. Then darken the right and bottom edges in the same way. You now have a basic block of marble. You now need some instruments, mark off a series of boxes (just a point at each corner will do, then highlight the top and left if you want the box to stick out or the right and bottom if you want it to look as if it sinks in. Shade the opposite sides and it is almost finished. Select a range of colours that contrast well with the basic marble colours, red through yellow is quite good. Then draw your movement icons and anything else you might need. I have included one example on my latest PD disk (to be reviewed next issue - Ed), and all of the data files to work with it.

Another tip to make your border look better: For the energy meter, create three instruments (all verticle bars in my case) then colour one a dark tone, one a middle tone and one a light tone. Place the darkest one in first then place the middle one over the top leaving two pixels of the darker one showing either side. Then do the same with the lighter one but only leaving one pixel. You now have a 3D looking bar. In my border I drew a beaker around the energy bar to look like a cup of blood. Make sure that all three instruments have the same variable and have the same length (100 or so). In my data files, if you shoot the floor, your energy will be reduced.

\*\*\*\*\*

## PROBLEM SECTION

"I have a couple of problems. Regarding the opening screen where there is a picture of the words 3D Construction Kit in three dimensional form, on my VGA monitor it appears in black and white whereas on the video the logo was seen to be in colour. When I choose quit, instead of returning to the DOS environment I am again given the screen where the user is prompted to enter a number from the languages. Whatever response I supply the computer "hangs up" on me."

*Neil Broom*, Hull - PC

The screen you saw on the video is the Amiga version and, as far as I am aware, is the only version that the logo screen appears in colour. The problem with the QUIT on the PC version has been noted and the programmers are going to fix this in the next version of the Kit. I am sorry that PC owners have had to cope with this annoying problem. I'll keep everyone informed when the programmers have a fix.....Mandy

"What do I have to do to specify camera's 1 to 5? However I try I can never get back to their positions, obviously some stupidly easy action I am doing incorrectly."

*John Hayward*, Croydon - 16 BIT

You cannot use Cameras in the Test Mode or within Stand-alone games but to use them within Edit Mode you should position yourself with the view that you wish the camera to hold, then select the MODE icon and flip through the different modes to the Camera you need. Then use the RIGHT mouse button on this to set the camera.....Mandy

"I have written an energy bar routine but my problem is that if you destroy the "baddie" while you are within its sensor range the energy bar routine keeps draining the energy off the player. Please could

you find a way to stop the energy draining even if you kill the baddie? The conditions I used are as follows:

CONDITION 1:

```
IFSENSED 4
THEN
CALL 1
ELSE
ENDIF
```

PROCEDURE 1:

```
ADDV 50 1
CMPV 150 1
IFGT
THEN
ADDV 25 2
CMPV 100 2
IFGT
THEN
ADDV 10 3
CMPV 50 3
IFGT
THEN
ADDV 5 4
CMPV 25 4
IFGT
THEN
ADDV 2 5
CMPV 12 5
IFGT
THEN
ADDV 1 6
CMPV 184 6
IFEQ
THEN
ENDGAME
END
```

The energy bar is 23 characters long and is controlled by variable 6."  
*Tom Reilly*, Liverpool - SPECTRUM

The only problem that I can see is that you need to add a condition to reset the variable if the "baddie" is shot. This can be done in a number of ways. Assuming that your "baddie" is the sensor itself you could reset your variable, or set it to hold a different value as soon as the sensor is shot and becomes invisible. In the 8 Bit versions of the Kit you will note that there is no command such as IFINVIS (which I must admit I really miss), however, the manual does explain how to check if an object is invisible or not and to act on a FALSE outcome, such as:

```
IFVIS 4
THEN
ELSE
SETV 0 6
ENDIF
```

The 8 Bit FCL has some advantages over the 16/32 Bit language but the lack of IFINVIS is a disadvantage, however, the programmers have provided alternative methods of reaching the results you need wherever possible.....Mandy

## NEWS

### NEW PD CLUB FOR AMIGA

One of our members, Peter J R Laws, informs me that he is setting up a PD CLUB for Amiga users which will cost £7.50 for UK members and £13.50 for Europe. For this Peter tells me you get a bi-monthly newsletter with TIPS and POKES (for games and PD) and will also contain Letters, Adverts, Reviews of PD disks, Mail order etc. It will be around 14 pages and A4 size. Not only that but Peter tells me that there will be 2 FREE PD GAMES WITH EVERY ISSUE! Sounds good to me. For more information and to get a PD list why not write to Peter at Aztec PD, 60 Standard Road, Enfield, Middx, EN3 6DP. Telephone/Fax 0992 717824 between 9am and 8pm.

### MORE ON-LINE NEWS

If you have a modem then why not take advantage of it to get more information and help about the 3D Kit? There are three sources that I know of where you can log on for Kit info. The first of these is: 3D Construction Kit CIX Conference - this is run by one of the programmers, Paul Gregory, and is open to all registered users and only registered users can gain access to it. Anyone wishing to gain access should simply MAIL Paul under the username DIMENSION with your registration number and as soon as possible Paul will contact me for confirmation of registration and you should receive a reply in your MAIL basket within a few days. The conference is used by people wishing to contact other users to exchange ideas, environments, FCL programs etc. The conference is regularly moderated by Incentive so any problems and queries could be answered by them. The conference also has a filelist so if you have an interesting environment or object or anything relating to 3D Kit you wish to share, simply upload it to the conference.

As I mentioned in the last Newsletter, Chris King has set up a section of his BBS devoted entirely to 3D Kit. Chris is already preparing a filelist of material for downloading but would like quite a lot more so please, if you have any routines etc that you would like to share with other users then why not upload them to Chris, or have a word with him about it first to find out more details. There is also quite a lot more to interest you on Chris' BBS which is called KING OF THE CASTLE. Go on, why not give him a ring on (0233) 620228 and see what it is all about. I understand that the board is now open 24 hours!

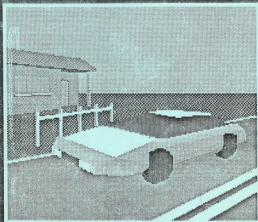
Don't forget that Marco Grubert also has a 3D Kit area on his BBS in Germany. This means that there will be lots of 3D Kit information available easily to everyone. The board is called ALPHA-BOX-BERLIN and the telephone number is (030) 744 71 06 and is open between 7am and 1am. Sysops are Prochnow, Cosysop Crossfire (that's Marco!).

If you log on then it might be a good idea to have your latest issue of the Newsletter handy as you might have to answer a couple of questions just to make sure that you are a bone fide member of the User Group.

If anyone has any information they would like to pass on to other members such as clubs, bulletin boards etc then please write in and I will gladly print details here for you. Obviously preference will be given to items which are specifically 3D Kit related but if there is room I will print other items as well.....Mandy



NEWSLETTER Aug/Sept 1991  
Issue No. 2



3D

CONSTRUCTION  
KIT USERS GROUP

3D

CONSTRUCTION  
KIT USERS GROUP